

REMARKS

Reconsideration and allowance are respectfully requested.

The amendments proposed in this Response particularly point out and distinctly claim the subject matter of the invention. The subject matter of cancelled claim 23 has been added to claim 22. No new matter has been added. Entry and allowance are requested.

Claims 1-23 are patentable under 35 U.S.C. 103(a) over US Patent 5,524,253 (Pham) and US Patent 6,640,255 (Snyder) because neither reference discloses the claimed system nor is either reference directed to the problem of sharing objects over a network.

The present invention addresses and provides a solution to a long-standing problem faced by program developers working in object-oriented programming of writing networking instructions that manage the sharing of classes of objects. This is a time-consuming task and involves extensive testing. The present invention describes a unique higher-level programming language that can be used to describe shared objects wherein network instructions are embedded within the object classes. Thus additional network-specific instructions are not needed. The invention relates to object-oriented programming defined in the present claims.

Pham relates to a system for integrating applications that are running on different platforms and written in different languages. According to Pham this problem can arise where

processes have been automated one at a time over a period of years resulting in the need to connect machines to each other (column 1, lines 23-42). Pham therefore describes a software tool that "translates" messages and data that need to pass from one machine to another (column 5, lines 11-17).

Pham does not discuss object-oriented programming and thus, Pham does not disclose memory means that "is configured to store program instructions for describing objects to be shared over a network by a plurality of network-connected terminals", nor that "said processing means is configured by said executable instructions set to manage the duplication of said described objects" as uniquely pointed out in claims 1, 11 and 21.

The Examiner interpretation of the word "objects" to include data files, and the "duplication of objects" to be analogous to the copying of such files, is in error. Claim 1 now defines "objects" to be the elements of object-oriented programming which should remove the Examiner's misunderstanding. Thus claims 1, 11 and 21 are not anticipated by Pham.

With regard to claim 23, Pham clearly does not disclose any of the required elements, such as "object class definition files", "object class description files", "linker", a "Data Definition Language compiler", a "Higher Level Programming Language compiler", a "Data Definition Language library", or "Higher Level Programming Language libraries". Thus claim 23 is not anticipated by Pham.

Pham is directed to the problem of creating a network

between differing machines and does not discuss object-oriented programming. The reference is directed to the problem of creating a network between automated systems, which is a totally different problem from that solved by the present invention. Thus Pham neither discloses nor renders obvious the amended independent claims.

Snyder relates to a method and apparatus for installing distributed objects on a distributed system. Distributed objects are used within object-oriented programming to provide a specific function. When a programmer requires this functionality he can make a call, through an interface, to the object, without himself understanding the object (column 1, lines 50 to 54). An additional advantage is that the object can be written in a single language but with multiple interfaces, each in a different language, allowing a programmer working in a different language to call an object (column 5, lines 1 to 4).

Snyder addresses the problem that it is difficult to bring a non-distributed object onto a distributed object system. The problem is solved by including, in the distributed objects, wrapper classes that inherit object attributes through an inheritance relationship with a developer-written class of objects. This means that the programming code required for distributed objects to operate in the distributed object system is provided transparently (column 9, lines 8 - 24).

The fundamental difference between Snyder and the present invention is that Snyder is directed to distributed objects,

whereas the present application relates to duplicated objects. Duplicated objects are duplicated such that an instance of the object exists on each computer within a network as required. Distributed objects exist in only one location on a network and their functionality is distributed by means of interfaces. Thus these two types of objects are in fact opposite to each other, with the consequence that the problems in using each of them are completely dissimilar.

Thus Snyder does not disclose, as required by claims 1, 11 and 21, that "said processing means is configured by said executable instructions set to manage the duplication of said described objects", because Snyder does not relate to the duplication of objects. Similarly, claim 22 requires that the "program instructions are configured to describe objects to be shared by a plurality of network-connected terminals over a network". Snyder does not disclose the sharing of objects, rather, it relates to access to a single instance of an object. Thus, the independent claims are novel over Snyder.

Also, there can be no question of obviousness because the problems are so different. There is no reason why one of ordinary skill in the art should look to the field of distributed objects, which has at its heart the notion of accessing an object in a single location, for an answer to a problem with duplicating objects.

Any combination of Pham and Snyder would still not produce the invention. Since neither of them even contemplate the

duplicating of objects there is no reason why the combination of the two disclosures should lead one skilled in the art to suggest the present invention.

Therefore, all the independent claims are novel, inventive, and non-obvious over both Pham and Snyder, and also over the combination of them, since neither addresses the same or even a similar problem to the present application. The dependent claims are also allowable.

Nothing in the references, either singly or in combination, teaches or suggests the claimed features. Therefore, the references cannot anticipate nor render obvious the present invention as claimed.

Since Applicant has presented a novel, unique and non-obvious invention, reconsideration and allowance are respectfully requested.

Respectfully,

A handwritten signature in cursive script, appearing to read "JC Wray", is written over the typed name of James C. Wray.

James C. Wray, Reg. No. 22,693
Meera P. Narasimhan, Reg. No. 40,252
1493 Chain Bridge Road, Suite 300
McLean, Virginia 22101
Tel: (703) 442-4800
Fax: (703) 448-7397

May 28, 2004